UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/666,032	09/18/2003	Eric Lawrence Barsness	ROC920030264US1	7940

46296 7590 03/31/2008

MARTIN & ASSOCIATES, LLC
P.O. BOX 548
CARTHAGE, MO 64836-0548

EXAMINER	
MITCHELL, JASON D	

ART UNIT	PAPER NUMBER
2193	

MAIL DATE	DELIVERY MODE
03/31/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/666,032
Filing Date: September 18, 2003
Appellant(s): BARSNESS ET AL.

Derek P. Martin (reg. # 36,595)
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 1/2/08 appealing from the Office action mailed 8/3/07.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

Haban, Weigel "Global Events and Global Breakpoints in Distributed Systems", 1988, IEEE INACS pp. 166-175.

US Re. 36,852 Heinen 09/2000

US 6,083,281 Diec et al. 07/2000

"Microsoft Computer Dictionary 5th edition" 2002 Microsoft, pg. 547.

Wikipedia "Process (computing)" available at

<http://en.wikipedia.org/wiki/process_%28computing%29>

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claims 8, 17 and 28 are rejected under 35 U.S.C. 103(a) as being unpatentable over "Global Events and Global Breakpoints in Distributed Systems" by Haban and Weigel (Haban) in view of US Re. 36,852 to Heinen, Jr. (Heinen).

Claims 9, 18 and 29 are rejected under 35 U.S.C. 103(a) as being unpatentable over "Global Events and Global Breakpoints in Distributed Systems" by Haban and Weigel (Haban) in view of US Re. 36,852 to Heinen, Jr. (Heinen) further in view of US 6,083,281 to Diec et al. (Diec).

Regarding Claims 8, 17 and 28: Haban discloses an apparatus comprising:

at least one processor (pg. 166, col. 1, par. 2 "processors");

a memory coupled to the at least one processor (pg. 166, col. 1, par. 2 "memory");

a first job residing in the memory and executed by the at least one processor (pg. 166, col. 1, par. 2 "multiple process running on multiple processors");

a second job residing in the memory and executed by the at least one processor (pg. 166, col. 1, par. 2 "multiple process running on multiple processors");

an inter-job breakpoint mechanism that detects at least one condition in the first job and, responds thereto by triggering an associated action (pg. 173, col. 2, par. 2 "the global event is satisfied and the action associated with the satisfaction is performed"),

Haban does not explicitly disclose the response comprises modifying a program variable in a second job, but does disclose sending a debug message to another job (pg. 173, col. 2, par. 2 "The local debuggers … exchange information among each other").

Heinen teaches a message that modifies a program variable in a second job (col. 7, line 30-32 "DEPOSIT – a message requesting that data forming part of the message be deposited in the memory of the specific job or process").

It would have been obvious to one of ordinary skill in the art at the time the invention was made to perform Heinen's "DEPOSIT" command (col. 7, line 30-32 "DEPOSIT – a message requesting that data forming part of the message be deposited in the memory of the specific job or process") as the "action" triggered by Haban's breakpoint (pg. 173, col. 2, par. 2 "the global event is satisfied and the action associated

with the satisfaction is performed"). Those of ordinary skill in the art would have been motivated to make such a change in order to provide specific known and useful debugging action (Heinen col. 7, line 30-32 "DEPOSIT") to a user of Haban's system. Such a combination would produce only the expected results (i.e. setting a variable / placing data in memory of a second job in response to detecting a condition).

Regarding Claims 9, 18 and 29: The rejections of claims 8, 17, 28 are incorporated, respectively; further, Haban discloses a response comprising outputting a debug message to a second job (pg. 173, col. 2, par. 2 "The local debuggers … exchange information among each other"). However, the Haban-Heinen combination does not explicitly disclose the response outputs a debug message to a second job's output.

Diec teaches a response comprising outputting a debug message to a second job's output (col. 2, lines 1-5 "issuing a message to another software object to trigger generation of tracing data"; col. 9, lines 57-59 "It receives messages ... and proceeds in channeling the entry into the logfile corresponding to the logfile identifier")

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Haban-Heinen and Diec in order to provide means for debugging distributed processes (Haban Title "Global Breakpoints in Distributed Systems"; Heinen Abstract "debugging … jobs or processes running on one or more remote units"; Diec col. 2, lines 10-12 "provide a distributed data network …

that has a tracing capability"). Those of ordinary skill in the art would have been motivated to make such a change in order to provide specific known and useful debugging action (Diec col. 2, lines 1-5 "issuing a message to another software object to trigger generation of tracing data") to a user of Haban's system. Such a combination would produce only the expected results (i.e. triggering generation of tracing data by a second job in response to detecting a condition)

(10) Response to Argument

Claims 8, 17 and 28

In the first par. on pg. 5, the appellants state:

> … The examiner then states Heinen teaches a message that modifies a program variable in a second job. The cited language of Heinen states:
>
> ... DEPOSIT - a message requesting that data forming part of the message be deposited in the memory of the specified job or process; …
>
> The examiner's mapping of Heinen on claim 8 fails for two reasons. First, the express teachings of Heinen do not teach or suggest modifying a program variable. The **depositing of information in the memory of a specified job or process does not imply modifying a program variable**. Second, **the DEPOSIT command in Heinen is a user command** as specified at col. 7 line 28 of Heinen. A DEPOSIT command thus allows a user to specify what data is deposited in the memory of a specified job or process. **The DEPOSIT command of Heinen thus expressly teaches away from the limitations in claim 8**, which states:
>
> .. an inter-job breakpoint mechanism that detects at least one condition in the first job and, in response thereto, modifies a program variable in the second job. (emphasis added)

The examiner respectfully disagrees. First the Microsoft Computer Dictionary 5th edition states:

> variable n. In programming, a named storage location capable of containing data that can be modified during program execution.

Secondly, Haban teaches examples of breakpoints wherein variables are consulted and manipulated in order to send a message (see pg. 166, Introduction and pg. 2.1, Primitive Events). Those of ordinary skill in the art would have recognized that manipulation and monitoring of variables by a process that execute separately and do not share memory and thus relays related messages between one another, to illustrate modify of a program variable by at least (1) the definition of a variable or the (2) functions or instructions of the process to the variable (i.e. assigning variables function Table 1).

In addition, it is noted that the claim does not include language indicating the modification be done 'automatically' or by a 'non-user action'. An inter-job breakpoint mechanism does not, in and of itself, mean it operates independently or without any assistance from the user. The claim language as written does not narrow the scope to either software or hardware, to manual operation or automated operation. Any such narrow interpretation of such would be read into the claims in violation of M.P.E.P. 2111 wherein the claims are interpreted based on the specification without reading in limitations from the specification. This is further supported by the Appellants statements made in the specification on page 15, lines 8-10 wherein the claims are not limited to a specific example. Thus, Heinen's disclosure of a user command does not 'teach away' from the claim since the claims are broad in scope and can be read in multiple manners. For the sake of providing a complete analysis, the examiner further notes that while Heinen's DEPOSIT command is preformed in response to a "user action" (i.e. the

entering of a command), the combination as applied would enable the DEPOSIT command to be performed in response to Haban's inter-job breakpoint detecting a condition. Thus the rejection is appropriate even assuming the appellants' inappropriately narrow interpretation of the claim language was permissible. Changing the 'triggering' mechanism of Heinen's DEPOSIT command as discussed would have been well within the skill of the ordinary artisan at the time to relay messages based on triggered breakpoints by a software mechanism (col. 7, lines 23-30).

In the par. bridging pp. 5-6, the appellants state:

> The modifying of a program variable in the second job as expressly recited in claim 8 is performed by the inter-job breakpoint mechanism in response to the inter-job breakpoint mechanism detecting at least one condition in the first job. Providing a user DEPOSIT command as taught in Heinen does not read on a software mechanism that modifies a program variable in the second job in response to the software mechanism detecting at least one condition in the first job. Because neither Haban, Heinen, nor their combination teach or suggest an inter-job breakpoint mechanism that detects at least one condition in the first job and, in response thereto, modifies a program variable in the second job, claim 8 is allowable over the combination of Haban and Heinen.

The examiner respectfully disagrees. As discussed above and in the rejection, Haban discloses a mechanism that triggers an action in response to "detecting at least one condition in a first job" (pg. 173, col. 2, par. 2 "the global event is satisfied and the action associated with the satisfaction is performed"). Heinen discloses a mechanism that modifies a program variable in a second job based on a triggering condition in another job (col. 7, line 30-32 "DEPOSIT – a message requesting that data forming part of the message be deposited in the memory of the specific job or process"). Those of ordinary skill in the art would have recognized that using a command similar to Heinen's

DEPOSIT command as the action discussed in Haban would provide a useful improvement of Haban's global breakpoints. Since the mechanism in the claims provide no direct indication of what it is, the Examiner believes his interpretation discussed herein are correct.

Starting in the second to last par. on pg. 6 the appellants state:

> Appellant respectfully asserts the examiner's rejection based on the combination of Heinen and Haban is in error. **The only motivation for combining Heinen and Haban** in the manner proposed by the examiner **is through the [use] of impermissible hindsight reconstruction**. Absent the teachings in appellant's claims, one of ordinary skill in the art would not be motivated to convert the user DEPOSIT command in Heinen into an automated command performed by a software process as recited in claim 8. …
>
> **The examiner has provided a very general motivation to combine Haban and Heinen, namely "to provide means for debugging distributed processes". This motivation is a general motivation that is present in both references**. This general motivation does not provide motivation for combining Haban and Heinen in the manner claimed
> (emphasis added)

The examiner respectfully disagrees. It is not clear if the appellants believe the motivation to combine came only from appellants own teachings or through a 'general motivation' expressed in the cited references. Regardless, Haban discloses a global breakpoint which triggers an unspecified action (pg. 173, col. 2, par. 2 "the global event is satisfied and the action associated with the satisfaction is performed"). Heinen recognized the utility of setting a variable in another job or process and implemented a command to do so (col. 7, line 30-32 "DEPOSIT – a message requesting that data forming part of the message be deposited in the memory of the specific job or process"). As stated in the rejection and above, it would have been obvious to use Heinen's

DEPOSIT command in conjunction with Haban's global breakpoints, thus providing additional utility to Haban's system. Thus it should be clear the motivation is not taken strictly from the appellants' disclosure and, however general, would still have provided those of ordinary skill in the art with a reason to make the asserted change.

Claims 9, 18 and 29

In the first par. on pg. 8, the appellants state:

> … In the rejection, the examiner admits the Haban-Heinen combination does not explicitly disclose the response outputs a debug message to a second job's output. The examiner then states Diec teaches these limitations, citing col. 2 lines 1-5 of Diec, which recites: "issuing a message to another software object to trigger generation of tracing data." The examiner's mapping of Diec on the limitations in claim 9, 18 and 29 is in error because issuing a message to another software object as taught in Diec does not read on outputting a debug message to the second job's output as recited in claims 9, 18 and 29. While issuing a message to another software object to trigger generation of tracing data may result in generating output from a job, the message is not in the output from the job. As a result, claims 9, 18 and 29 are allowable over the combination of Haban, Heinen and Diec.

The examiner respectfully disagrees. The limitation in question reads "The inter-job breakpoint … outputs a debug message to the second job's output." This does not indicate specific steps which taken to output 'debug message'; and further does not specify what form the debug message must take or where the message comes from. In view of this breadth, Diec's teachings meet the claimed limitation. Specifically, trace data (Diec col. 2, lines 1-5 "generation of tracing data") is known to be useful for debugging (See Diec, Abstract, "Tracing software entities is important for software

developers to permit the quick localization of errors and hence facilitate the debugging process.") and thus constitutes the broadly claimed 'debug message'. Diec's trace data is clearly relayed from software components to software component until the trace information is written to "a file output" (col. 2, lines 1-5 "issuing a message to another software object to trigger generation of tracing data"; col. 9, lines 57-59 "It receives messages ... and proceeds in channeling the entry into the logfile corresponding to the logfile identifier"; see also col. 6, lines 24-26; col. 7, lines 23-31; col. 7, lines 46-60; and col. 8, lines 4-7). Accordingly, Diec's teachings, in combination with the disclosure of Haban read on the broad interpretation of the limitations, specifically Haban's inter-job breakpoint detects met conditions and outputs (by sending Diec's message) a debug message (Diec's trace data) to a second job (Haban's second job) for later storage in an output file system.

Additionally, it is noted that processes generally store in memory a process specific 'output buffer' (see e.g. Wikipedia "Process (computing)" pg. 2 "in general, a computer system process consists of … Memory … which includes … process-specific data (input and output)"). With this in mind it can reasonably be stated that Heinen's DEPOSIT command (col. 7, line 30-32 "DE POSIT – a message requesting that data forming part of the message be deposited in the memory of the specific job or process") could be used by those of ordinary skill in the art to 'deposit' data into such an output buffer (i.e. "requesting that data … be deposited in the memory") thus meeting the

claimed limitation (i.e. “outputs a message to the second job’s output”) while only relying on the teachings of the Diec reference for the message being a debug message.

In regards to the arguments regarding the motivation for combination, Appellant states that the examiner has provided a very general motivation to combine Haban, Heinen, and Diec, namely "to provide means for debugging distributed processes" and does not provide motivation for combining Haban, Heinen and Diec in the manner claimed. Appellant further states that nowhere within the references or available to one of ordinary skill in the art to make the combination. The examiner disagrees. Each of the references applied all detail the concept of the independent claims of using a mechanism that detects at least one condition in the first job or component and in response to such send a message, for example an update message, to a second job or component. Diec is no different in that his purpose for sending a message is to initiate the debugging of the distributed application. It states such in the abstract of Diec. Therefore, the motivation for sending a message to another job or component comes directly from the references. The examiner believes there is no hindsight being applied to the interpretation of the references and that the references are properly combined and meet the language of the claims.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner’s answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

/Jason Mitchell/

Jason Mitchell

Conferees:

/Lewis A. Bullock, Jr./

Supervisory Patent Examiner, Art Unit 2193

Tuan Dam

/Tuan Q. Dam/

Supervisory Patent Examiner, Art Unit 2192